Machine Learning (CSE 446):
Practical issues: optimization and learning

Sham M Kakade

© 2018

University of Washington
cse446-staff@cs.washington.edu
Announcements

- Midterm summary:
  - stats: 71.5 std: 18
  - Office hours today: 1:15-2:30 (No office hours on Monday)

- Monday: John Thickstun guest lecture

- Grading:
  - HW: 60%
  - Midterm: 15%
  - Final: 25%

- HW3 posted
  - will be periodically updated for typos/clarifications
  - extra credit posted soon

- Today:
  - Midterm review
  - GD/SGD: practical issues

\[
(x_1, x_2) \rightarrow (x_1, x_2, x_1^2, x_2^2, x_1 x_2)
\]
Midterm
What is a good model of this distribution?
What is a good model of this distribution?

“A mixture of Gaussians”
Midterm Q4: scratch space

\[ \| w \|_2^2 \geq \frac{1}{\| x \|_2} \]

for all \( y \),

when it has a solution,

\[ y_0 \cdot (w \cdot x) \geq 1 \]

\( \iff \) linearly separable

+ -
- +
+ -
Midterm: scratch space
Midterm Q5: scratch space

- $(1, 0)$

- $4 - 4 = 0$

- $\frac{1}{4} (x^2 + 4^2 + 0 + 0) = 8$

- $(4, 0), (-4, 0), (0, 0), (0, 0)$

- $\frac{1}{4} (0 + 0 + 1 + 1) = \frac{1}{2}$
Midterm: scratch space
Today
The “general” Loss Minimization Problem

\[ w^* = \arg\min_w \frac{1}{N} \sum_{n=1}^{N} \ell(x_n, y_n, w) + R(w) \]

How do we run GD? SGD? Which one to use?

How do run them?
Our running example

\[
\arg\min_w \left\{ \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} (y_n - w \cdot x_n)^2 + \frac{1}{2} \lambda \|w\|^2 \right\}
\]

- GD? SGD?

- Note we are computing an average. What is a crude way to estimate an average?

Will it converge?
How does GD behave? A 1-dim example

\[ \min_{\omega} f(\omega) \]

\[ f(\omega) = \frac{1}{2} \omega^T \omega \]

\[ Df(\omega) = \omega \]

\[ \omega \in \Omega \]

\[ \Omega = \omega - \epsilon \Omega \]
GD: How do we set the step sizes?

- Theory:
  - square loss:
  - more generally:

- Practice:
  - square loss:
  - more generally:

- Do we decay the stepsize?

try things out to get it stable
SGD for the square loss

**Data:** step sizes \( \langle \eta^{(1)}, \ldots, \eta^{(K)} \rangle \)

**Result:** parameter \( w \)

initialize: \( w^{(0)} = 0 \);

for \( k \in \{1, \ldots, K\} \) do

\( n \sim \text{Uniform}\{1, \ldots, N\} \);

\[ w^{(k)} = w^{(k-1)} + \eta^{(k)} \left( y_n - w^{(k-1)} \cdot x_n \right) x_n; \]

end

return \( w^{(K)} \);

**Algorithm 1:** SGD
SGD for the square loss

Data: step sizes $\langle \eta^{(1)}, \ldots, \eta^{(K)} \rangle$

Result: parameter $w$

initialize: $w^{(0)} = 0$

for $k \in \{1, \ldots, K\}$ do
  $n \sim \text{Uniform}\{1, \ldots, N\}$;
  $w^{(k)} = w^{(k-1)} + \eta^{(k)} (y_n - w^{(k-1)} \cdot x_n) x_n$;
end

return $w^{(K)}$;

Algorithm 2: SGD

- where did the $N$ go?
- regularization?
- minibatching?
SGD: How do we set the step sizes?

- Theory:

- Practice:
  - How do start it?
  - When do we decay it?
Stochastic Gradient Descent: Convergence

\[ w^* = \arg\min_w \frac{1}{N} \sum_{n=1}^{N} \ell_n(w) \]

- \( w^{(k)} \): our parameter after \( k \) updates.
- Thm: Suppose \( \ell(\cdot) \) is convex (and satisfies mild regularity conditions). There is a decreasing sequence of step sizes \( \eta^{(k)} \) so that our function value, \( F(w^{(k)}) \), converges to the minimal function value, \( F(w^*) \).
- GD vs SGD: we need to turn down our step sizes over time!
Making features: scratch space